Introduction

This guide provides step-by-step instructions for setting up an online Bitbucket repository. It's designed specifically for FSU students using Windows machines who will be utilizing repositories for various technology courses, but are unfamiliar with Git or Bitbucket.

By following this guide you will: save time, understand new concepts, avoid common errors, and develop a solid foundation for managing code repositories, which is a crucial skill for future coursework and professional projects.

List of required materials

This guide assumes you are using a standard computer setup with a keyboard and mouse, and that you have a basic understanding of using a file explorer and browser.

- Windows computer or Laptop with internet access.
- An internet browser (Chrome, Edge, Firefox, etc.).
- An FSU email address.

Definitions

Before we begin, here's a list of definitions for terms featured throughout this guide:

- **Repository**: Also known as a code repository, or simply "repo" in the tech community, it is a storage location for code and other software documentation assets.
- Local Repository: A folder stored locally on a computer that contains code, files, and other information.
- **Online Repository**: A digital storage space that contains code, files, and other information.
- Version Control System: A Version Control System is a software tool that helps software developers manage and keep track of code over time. They track every modification made to the code. If a mistake is made, developers can compare earlier versions of the code to help fix the mistake.
- Git: A free, open-source Version Control System.
- **GitBash**: GitBash is a Windows application that lets you use Git commands in a command terminal. "Bash" stands for Bourne Again Shell, which is a terminal used to interact with an operating system using text commands.
- **Bitbucket**: Bitbucket is a Git-Based code repository, which means it uses the **Git Version Control System** to track changes in code, collaborate with other developers, and manage project versions. This is an example of an **Online Repository**.

• **SSH Key**: Essentially a 'digital key' that grants permission to users to access online information. In the context of this guide, they'll be used to access your **Online Repository**.

Guide Overview

- 1. Download and set up GitBash
- Install the Gitbash application to enable Git commands on your machine.

2. Set up your Local Repository

• Create a folder on your computer where you'll store your project files and initialize it with Git.

3. Create a Bitbucket Account and Set Up an Online Repository

- Register a free account with Bitbucket using your FSU ID.
- Create an online repository through Bitbucket.

4. Create an SSH key.

• Generate SSH keys locally, then enter them in Bitbucket.

5. Connect your local and online repositories.

• Use Git commands in the command terminal to connect your repositories.

Part 1: Download and set up GitBash

First, we'll be downloading GitBash. GitBash will allow you to connect your repositories and issue commands through a terminal window.

- 1. Click the following link to navigate to the Git for Windows Download site: <u>Git for</u> <u>Windows Download</u>.
- 2. Select "**Download**" as shown in the image below:



Figure 1 - Git for Windows: Download Circled

- 3. Once the install has finished, select it through the Download section of your browser, or locate the file in your downloads folder and double-click the icon to start the application.
- 4. A pop-up will appear asking if you'd like this application to make changes to your device. Select **"Yes."**
- 5. We'll be using the default configuration of Git, so select "Next" until you get to "Install," then select "Finish" to complete the installation.

Next, we're going to add your username and email through the GitBash terminal. We'll be using your FSU ID and Email.

1. Now that the install is complete, select the Windows key and type "Git Bash" then press enter. You should see something like this:

MINGW64:/c/Users/user	_	\times
user@DESKTOP_EXAMPLE MINGW64 ~		<u> </u>
\$ [
		\sim

Figure 2 - GitBash Terminal

- 2. Enter the following commands into the terminal, entering your FSU ID and FSU Email where indicated:
 - a. git config --global user.name "FSU ID"
 - b. git config --global user.email "FSU Email"

You've now set up your username and ID. These details will be associated with any code changes you later make through Git.

To verify that Git is working correctly, enter the following command into the GitBash terminal to display the current version of Git:

• git --version

As of 1/25/2025, the latest version is Version 2.47.1.windows.2

Now that Git has been installed, we can move on to Part 2.

Part 2: Set up your Local Repository

Now that GitBash has been installed, we can go ahead and create your local repository. This is where you'll store all of your code, and eventually push it to your online repository.

We'll be placing it in your user folder.

- 1. Open your file explorer.
- 2. Using the Side Bar (also known as the Navigation Pane, shown below), click on "This PC"



- 3. In the main window, double-click the "(C:)" drive. This is usually where the operating system and user folders are stored.
- 4. Inside the C: drive, you'll see several folders. Look for a folder named "Users" and double-click to open it.
- 5. Inside the User folder, locate the folder corresponding to your account name (usually your login name) and open it.

Now that you've navigated to the correct folder, we'll go ahead and create a file tree where you'll be able to create multiple repositories for different FSU classes.

- 1. First, right-click and create a new folder. Name it "repos"
- 2. Inside the "repos" folder, create a new folder named "classes". This folder will store repositories created for any class that requires it.
- 3. Open the "classes" folder. Create a new folder using the course pre-fix and course number of your class.

Here's an example: lis4381

Now that you've successfully created the folder for your local repository, now we have to initialize it through GitBash.

What does it mean to initialize a folder? When you initialize a folder through GitBash, your system will create a folder named ".git," where it will store all the information it needs to manage your code. From there, it will allow you to save your work and track changes.

First, since .git is a "hidden" file, we'll need to make sure you can see it in the File Explorer.

- 1. In the File Explorer, locate and select the "View" tab at the top of the application.
- 2. Make sure that "Hidden Items," which can be seen highlighted in the image below, is checked.

Image: Image	View					- □ × ^ (?
Navigation Details pane	Extra large icons Large icons Small icons Ist Image: Small icons Ist Image: Small icons Ist Image: Small icons Ist	Medium icons	Group by • M Add columns • Sort by • H Size all columns to fit	 ☐ Item check boxes ☑ File name extensions ☑ Hidden items 	Hide selected items	
Panes	Layout		Current view	Show/hide	e	
\leftarrow \rightarrow \checkmark \uparrow \square \rightarrow Th	is PC → Local Disk (C:) → Users →	> classes		✓ Č Search classes		م
📌 Quick access	▲ Name	Date n	nodified Type	Size		
	lis4331	1/22/2	025 2:21 PM File folder			
 OneDrive - Personal OneDrive - Attachments 	-					

Figure 3 - Hidden Items checkbox highlighted

Now that hidden items are displayed, lets initialize your local repository. We'll now navigate to the class folder you created through the GitBash terminal.

1. Open the GitBash terminal and enter the following command, typing the name of your class folder where the highlighted text is shown below:

cd repos/classes/{whatever class name you entered previously}

2. Press Enter.

GitBash should display your current directory's file path in the command line after your name, but if you're not sure if you are in the right folder, type the following command for the terminal to display it to you:

pwd

3. Once you've confirmed you're in the correct folder, enter the following command:

git init

4. Go back to the File Explorer and confirm that a brand new .git file is present.

I Image: Second s	1 hare View		- 🗆 ×
Navigation Details paper	ane Extra large icons Large icons	Sort Urrent view Show/bide	Options
$\leftarrow \rightarrow \checkmark \uparrow \square \checkmark$	Users > jason > repos > classes > lis4331	✓ C Search lis4331	م
 Quick access OneDrive - Person OneDrive - Tallah: This PC Network 	Name ^ .git assee Commu	Date modified Type 1/22/2025 10:51 PM File folder	Size

Figure 4 - .git now displays in File Explorer

You have successfully created and initialized your local repository using Git and can now move on to Part 3.

Part 3: Create a Bitbucket Account and Set Up an Online Repository

Now that you've got your local repository created, we can start setting up its online counter-part. First, we'll navigate to the Bitbucket website and set up a new account.

- 1. Navigate to the Bitbucket website by clicking this link (<u>Bitbucket</u>) and selecting "Get it free" in the top right corner of the site.
- 2. It should take you to a screen that says, "Log in to continue." Select "Create an account" located towards the bottom of the screen, which can be seen highlighted in the image below.

ATLASSIAN
Log in to continue
Enter your email
Remember me
Continue
Or continue with:
G Google
Microsoft
🗳 Apple
Slack
Can't log in? • Create an account
Content of State of S

Figure 5 - Account Setup Screenshot

3. Enter your FSU email address and click "Sign Up." Enter the code sent to your email, then select "Verify."

- 4. Enter your name and create a password, then select "Continue."
- 5. Where it says "Your new workspace," enter something professional, but be aware, it has to be a unique name Bitbucket does not allow two workspaces with the same name to exist.
- 6. After it finishes setting up your workspace, open it. Once inside, click on "Create Repository"

Make sure you follow the next few steps closely, and only press "Create Repository" when indicated.

- 1. For both the "Project name" and "Repository name" fields, enter the same name you gave your local repository during Part 2, with the course pre-fix and course number (**Example: lis4381**).
- 2. For "Access Level" field, it depends on what your professor wants. Leaving it public means anyone can access, but making it private means you'll have to add your professor as a read-only user.
- 3. For dropdown fields "Include a README?" and "Include .gitignore?," select No for both.
- 4. For "Default branch name," your professor may want you to enter "master." This can be changed later, so it's alright to skip it if you're not sure.
- 5. Select "Advanced settings" and select Java for the "Language" dropdown field.

Create a new re	pository	Import repository
Workspace	E	
Project*	■ lis4381	
Repository name*	lis4381	
Access level	 Private repository Uncheck to make this repository public. P typically contain open-source code and ca 	ublic repositories an be viewed by anyone.
Include a README?	No	
Default branch name	master	
Include .gitignore?	No	
✓ Advanced settings		
Description		<i>li</i>
Language	Java x 🗸	
	Create rep	cository Cancel

6. Once you've verified that your screen reflects the same options shown above, select "Create repository"

Excellent. Now that you've created your repository, we can go ahead and create your SSH keys.

Part 4: Create your public and private SSH Keys

1. Make sure that you have an SSH installed with Git by typing the following command into the GitBash terminal:

ssh -V

2. Open your GitBash terminal, type the following command to navigate to your home or user directory:

cd ~

3. Generate an SSH key pair using the ssh-keygen command, replacing the text, **FSU Email**, with your FSU email:

ssh-keygen -t ed25519 -b 4096 -C "FSU Email" -f ssh-key

- 4. The system will prompt you with a password. It's your choice to enter one or leave it empty. If you input a password, you will be prompted for this password every time an SSH is used, such as using Git to contact Bitbucket (pushing, pulling, fetching). Providing a password will prevent other users with access to your device from using your keys.
- 5. Once complete, two files will be created in your user folder with the names:
 - a. ssh-key.pub Your public key
 - b. ssh-key Your private key

Now that you have your public and private keys, we'll need to add your public key to your Bitbucket account in order to connect.

1. Go to your Bitbucket workspace and locate the settings cog, as shown highlighted in the following image.

🔢 🔽 Bitbucket	Your work	Pull requests	Repositories	Projects More 🗸	Create ~				Q. Search	🔹 📀 🔵
			Recent repo	ositories +					View all	
			Java lis4381	,						
			Pull request	ts		You have no open pull requests.				
			Bring mc See owners, linking them Get Comp	ore context to dependencies, and a to a central softwa hass free Learn r	y your code help resources for your repositories by re catalog.			al Co	mpass	
			Jira issues							
					Connect your commits, branches and	d pull requests to issues, and view, ec Connect Jira sites	t and transition issues from	m Bitbucket.		

Figure 6 - Showing the Settings Cog on the Bitbucket UI

2. In the dropdown list, select "Personal Bitbucket Settings" as highlighted in the image below.



3. Locate and select "SSH keys" in the lefthand side-bar, then select Add Key.

- 4. For the Label field, enter "Public"
- 5. Now, in your File Explorer, locate the public SSH file you generated previously, named "ssh-key.pub" in your user folder.
- 6. Right-click the file, select "Open with..." then select "Notepad," or any other text editor you would prefer.
- 7. Select all text in the file by pressing Ctrl + A, copy it with Ctrl + C, then go back to Bitbucket and paste it into the "Key" text field.
- 8. Select Save.

Now that you've added your public key, we'll be adding your private key through the GitBash terminal. This is important - You'll need to run the following commands every time you want to connect to your online repository.

- 1. Open GitBash.
- 2. Type the following command, **including the \$ at the front of eval**:

eval "\$(ssh-agent -s)"

You should see output like this (not exact, just an example): Agent pid 1234

This means that the agent is running. Now you'll need to add your private key to the SSH agent.

1. Type the following command:

ssh-add ~/ssh-key

If you previously created a passphrase, you'll be asked to enter it here.

Before proceeding to your repository, let's test the connection to Bitbucket using the following command:

ssh -T git@bitbucket.org

If successful, you'll see a message similar to the following:

authenticated (username)! You've successfully authenticated, but Bitbucket does not provide shell access.

If you run into an error with this, I suggest closing GitBash, then going back to the start of Part 4 and following the previous steps. If you're still having issues, I recommend using a search engine like Google to find solutions, or using an AI software like ChatGPT which will dynamically walk you through whatever the problem is.

Now, let's finally connect your local and online repositories.

1. Navigate to your repository by typing the following command, typing the name of your class folder where the highlighted text is shown below:

cd ~/repos/classes/whatever class name you entered previously

 Now, type the following command, replacing the yellow highlighted text with the name of your Bitbucket Workspace, and the green highlighted text with the name of your Bitbucket Repository

git remote add origin git@bitbucket.org:workspace/repository.git

3. Let's test the connection by typing the following command:

git remote -v

If you've successfully linked your Local Repository with your Online Repository on Bitbucket, GitBash will display something similar to the following:

origin git@bitbucket.org:<mark>workspace</mark>/repository.git (fetch) origin git@bitbucket.org:<mark>workspace</mark>/repository.git (push)

References

A brief overview of Bitbucket [Bitbucket Definition]

https://bitbucket.org/product/guides/getting-started/overview#a-brief-overview-ofbitbucket

Getting Git right [Git Definition]

https://www.atlassian.com/git/tutorials/what-is-version-control

Git Bash [GitBash Definition] <u>https://www.atlassian.com/git/tutorials/git-</u> <u>bash#:~:text=Git%20Bash%20is%20an%20application,operating%20system%20through</u> <u>%20written%20commands</u>

Git: Working with Local Repository vs Working with Remote Repository [Local Repository Definition] <u>https://levelup.gitconnected.com/git-working-with-local-repository-vs-working-with-remote-repository-b2ec00df9a2e</u>

What are Code Repositories? [Repository Definition] https://www.sonatype.com/resources/articles/what-are-code-repositories

What is Repo? [Online Repository Definition] <u>https://aws.amazon.com/what-</u> <u>is/repo/#:~:text=A%20repository%2C%20or%20repo%2C%20is,same%20project%20fro</u> <u>m%20any%20location</u>

Basic overview of SSH Keys [SSH Key Definition] https://www.ssh.com/academy/sshkeys#:~:text=SSH%20keys%20enable%20the%20automation,control%20who%20can%2 0access%20what.

Jason Westmark (2024) [Git for Windows Screenshot] Setting up a Bitbucket Online Repository

Jason Westmark (2024) [File Explorer Screenshot] Setting up a Bitbucket Online Repository Jason Westmark (2024) [Hidden Files Checkbox Screenshot] Setting up a Bitbucket Online Repository

Jason Westmark (2024) [.git Screenshot] Setting up a Bitbucket Online Repository

Jason Westmark (2024) [Creating a Bitbucket Account Screenshot] Setting up a Bitbucket Online Repository

Jason Westmark (2024) [Creating a Bitbucket Repository] Setting up a Bitbucket Online Repository

Jason Westmark (2024) [Bitbucket UI - Settings Cog] Setting up a Bitbucket Online Repository

Jason Westmark (2024) [Bitbucket Settings Dropdown] Setting up a Bitbucket Online Repository